UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Detección de relaciones causales entre noticias en medios en español

Álvaro Cortés Devesa
Tutor: Kostadin Koroutchev

Junio 2019

Detección de relaciones causales entre noticias en medios en español

AUTOR: Álvaro Cortés Devesa
TUTOR: Kostadin Koroutchev

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2019

Resumen

Las noticias dadas en artículos de diferentes medios raras veces tienen múltiples fuentes. Es habitual encontrar artículos cubriendo un evento, que sean redactados en distintos medios, sin que estos estén basados en experiencias personales o siquiera en fuentes propias. En los últimos tiempos se ha empezado a dudar de los medios y del origen de estas fuentes comunes entre ellos. Se ha empezado a hablar del fenómeno llamado *fake news* (popularizado en EEUU) como instrumento de manipulación, que hace que haya una mayor preocupación e interés por conocer la fuente y procedencia de las noticias que se exponen al público. Este es un tema de visible preocupación siendo unos de los centrales en redes sociales y blogs de opinión en el último año.

Es por ello por lo que este trabajo trata de establecer la relación entre los artículos de noticias que existen en los medios de comunicación utilizando métodos estadísticos, empíricos y de la teoría de información. Esto ha motivado la creación de un sistema con el que poder extraer artículos de principales medios en la web, clasificarlos por evento y construir una base para el estudio y análisis de estas relaciones. Como resultado final se ofrece un sistema capaz de reflejar qué medios comparten fuentes entre sí a la hora de escribir artículos sobre una misma noticia.

Palabras clave

Artículos, noticias, fuentes, noticias falsas, medios de comunicación, manipulación, población, opinión.

Abstract (English)

The articles covering news on different news media companies rarely have multiple sources. It is common to find articles covering an event, redacted by different media, not basing them on their own sources or the journalists' own experience. In recent times, a general concern has been raised among the public, awakening the disbelieve in the mainstream media companies. The fake news phenomenon is present nowadays as one of the main topics in social media and personal blogs. This idea makes people wonder how truthful and valid are the sources used by the media. It phenomenon is seen as a manipulation tool, making people want to know the source and origin of the news they are exposed to.

This is the reason why this Bachelor final thesis tries to establish the relationship between news articles using statistical, empirical methods and information theory. This has motivated the creation of a system capable of extracting articles from the mainstream media websites, classify them by event and build a base software to use for the study and analysis of these relationships. The result is a system to identify which articles covering the same event have been redacted using the same sources.

Keywords

News articles, sources, fake news, news media companies, trust, population manipulation, opinion

A Juan Carlos y Laura. El mero hecho de escribir vuestros nombres me hincha el pecho de felicidad. No creo que existan palabras que puedan describir lo afortunado que me siento de que seáis parte de mi vida.

A mi tutor Kostadin. Primero por darme ese ciego voto de confianza al aceptar a un alumno con trabajo full-time en el extranjero y sin conocerle ni tener referencias. Y más tarde por descubrir en él una persona que irradia pasión por el saber y lo transmite con conversaciones que atrapan.

Índice de contenidos

1.Introducción

1.1 Motivación

Las noticias raras veces tienen múltiples fuentes [1]. Lo habitual es que una noticia se redacte en distintos medios sin que esta esté basada en experiencia personal o fuente propia. No solo eso, se ha demostrado que las personas que más comparten y opinan en redes sociales son normalmente las menos capaces a la hora de reconocer artículos imparciales o de poca veracidad [2]. La gran actividad y publicación de artículos intentando influir en los lectores es capaz de visualizarse creando mapas de relaciones entre sitios web que cooperan para ello [3] y además, a causa de la preocupación que estos hechos levanta, se están investigando cada vez más métodos, no solo para detectar [5][6] sino para poder entender cómo se propagan este tipo de noticias falsas [4][8][10] y el impacto que tienen en comunidades especificas [7][9].

Por lo tanto, es de interés saber la fuente de una noticia. Este trabajo trata de establecer la relación entre las noticias que existen en los medios de comunicación utilizando métodos estadísticos, empíricos y de la teoría de información.

1.2 Objetivos

El objetivo de este trabajo de fin de grado es diseñar e implementar un sistema con el cual poder:

- Obtener y almacenar artículos de distintos medios de comunicación.
- Dado un evento, agrupar todos los artículos que lo cubran.
- Analizar la relación de los artículos dentro de cada grupo.

Este trabajo tiene a su vez como objetivo ser la herramienta base de posteriores trabajos centrados en el tercer y último punto listado.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Diseño. En este capítulo se da definición al sistema a implementar.
- Desarrollo. En él se habla de aquellas tecnologías elegidas para la implementación del trabajo, hablando de las características y elementos que justifican su papel en el diseño del sistema. Se explicarán los detalles técnicos a la hora de implementar este proyecto.
- Pruebas y Resultado final. Se hablarán de las pruebas que darán justificación a elecciones tomadas durante el desarrollo.
- Conclusiones y Trabajo futuro. Posibles mejoras para la herramienta e ideas para trabajos futuros tomando este como base.

2. Diseño

2.1 División de responsabilidades en módulos

El trabajo, como ya se ha indicado anteriormente, ha de cumplir tres objetivos:

A. Obtener y almacenar artículos de distintos medios de comunicación.
B. Dado un evento, agrupar todos los artículos que lo cubran.
C. Analizar la relación de los artículos dentro de cada grupo.

Inspirado en la filosofía Unix: "Haz programas que hagan solo una cosa y la hagan bien" [1] el sistema estará dividido en tres subprogramas o módulos. Cada uno de ellos se centra única y exclusivamente en conseguir uno de los objetivos, intentando ser lo más independientes posible entre ellos.

Los dos primeros objetivos son tareas concisas y bien definidas que pueden fácilmente seguir esta filosofía Unix. Ambos son módulos de Python independientes que hacen sus tareas y las hacen de forma efectiva.

El tercer objetivo es cubierto por un módulo que analiza ciertos aspectos de los artículos individualmente y los compara para estudiar las posibles relaciones. Este último módulo, basado en el resultado de los dos primeros, constituye en esencia una base que ampliar en futuros trabajos.

En la (Figura 3-1) se muestra a alto nivel cómo será el flujo o interacción entre ellos a través de la base de datos. A continuación, una breve descripción de cada módulo:

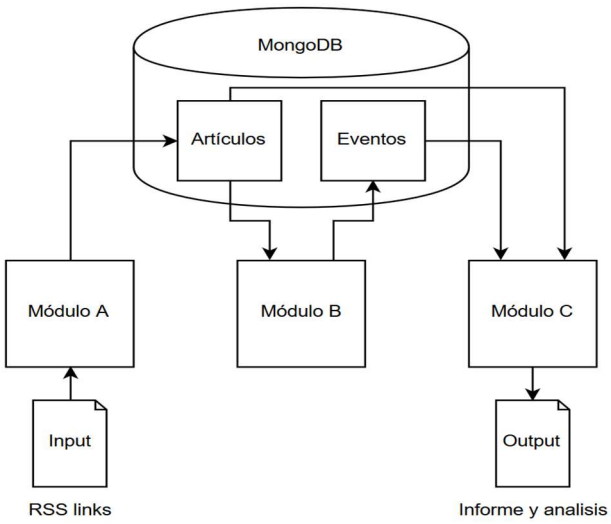

Figura 3-1: Esquema del sistema

2.1.1 Módulo A

Entrada

Archivo de texto con las URL de todos los feeds RSS que se quieran analizar. El archivo contendrá una URL por línea.

Salida

Colección de artículos de noticias en la base de datos.

2.1.1.1 Requisitos funcionales del módulo A

RF1. El módulo generará la salida de forma completamente automática.

Provista la entrada en forma de archivo de texto con una URL por línea, el módulo A será capaz de crear en la base de datos una colección de artículos sin intervención manual del usuario.

RF2. El módulo A será capaz de borrar y crear la base de datos.

Al iniciar la aplicación el módulo A tendrá la responsabilidad de limpiar la base de datos de previas ejecuciones. Es por eso por lo que el módulo tendrá que ser capaz de borrar la base de datos y crear una nueva.

RF3. El módulo A será capaz de crear una colección dentro de la base de datos.

El objetivo principal del módulo es la creación de una colección de artículos dentro de la base de datos, por lo que este tendrá que ser capaz de crear una colección dentro de la misma.

RF4. El módulo A creará los índices de texto MongoDB como es descrito en la sección 2.2 de este documento.

La funcionalidad de índices basados en campos de texto de MongoDB hace posible y sencillo cumplir los objetivos esenciales previamente mencionados en este mismo documento. Los índices creados harán posible la búsqueda de texto en la colección de artículos.

RF5. El formato de los documentos, en el caso de este módulo artículos de noticias, será como el descrito en la sección 2.2 de este documento.

Respetar este punto hace posible conectar posteriormente el módulo B, que espera como entrada una colección de documentos con este formato definido.

RF6. El módulo A será capaz de insertar documentos en la base de datos.

El objetivo principal del módulo es crear una colección en la base de datos. Es por ello por lo que es esencial que el módulo A sea capaz de insertar documentos dentro de la colección de artículos que más tarde utilizaran los módulos B y C.

RF7. El módulo no insertará en la base de datos dos artículos con la misma URL.

Es requisito no introducir en la colección resultante artículos con la misma URL, es decir, artículos que se ven listados en diferentes feeds RSS. Los medios en España suelen tener feeds RSS dedicados a diferentes secciones. Por ejemplo, esto hace que sea habitual encontrar el mismo artículo en la sección Portada y la sección Internacional.

2.1.2 Módulo B

Entrada

Colección de artículos de noticias en la base de datos creada previamente por el módulo A.

Salida

Colección de eventos en la base de datos.

2.1.2.1 Requisitos funcionales del módulo B

RF8. El módulo B generará la salida de forma completamente automática

Teniendo la colección de artículos en la base de datos producida por el módulo A, el módulo B será capaz de crear en la base de datos una colección de eventos sin intervención manual del usuario**.**

RF9. El módulo B será capaz de crear una colección dentro de la base de datos.

El objetivo principal del módulo B es la creación de una colección de eventos dentro de la base de datos, por lo que este tendrá que ser capaz de crear una colección dentro de la misma.

RF10. El módulo B será capaz de efectuar búsquedas en la base de datos.

Haciendo uso de la funcionalidad de índices basados en campos de texto de MongoDB configurada en la colección de artículos por el módulo A, el módulo B podrá hacer búsquedas de texto.

RF11. El formato de los documentos de la colección de eventos insertados en la base de datos será como el descrito en la sección 2.2 de este documento.

Respetar este punto hace posible conectar posteriormente el módulo C, que espera como entrada tanto una colección de artículos como una colección de eventos con los formatos definidos.

RF12. El módulo B será capaz de insertar documentos en la base de datos.

Para poder llevar a cabo su actividad, el módulo tendrá que ser capaz de insertar nuevos documentos en la base de datos para construir la colección de eventos que posteriormente será utilizada por el módulo C.

2.1.3 Módulo C

Entrada

Colección de artículos de noticias en la base de datos creada previamente por el módulo A y colección de eventos en la base de datos creada por el módulo B.

Salida

Breve análisis del evento y artículos, así como la representación de un árbol genealógico para el análisis de fuentes comunes entre artículos.

2.1.3.1 Requisitos funcionales del módulo C

RF13. El módulo C será capaz de efectuar búsquedas en la base de datos.

El módulo C tendrá que poder buscar en la base de datos en la colección de eventos aquellos que contienen más artículos. Podrá obtener dichos artículos de la colección de artículos haciendo búsqueda por el campo URL.

RF14. El módulo C generara la salida de forma completamente automática.

El módulo C dará su resultado de forma automática sin necesidad de intervención del usuario.

RF15. El módulo C será capaz de calcular el grado de similitud entre dos textos.

Para poder hacer un análisis de las fuentes de los artículos, el módulo C tendrá que ser capaz de comparar los textos de dichos artículos dando como resultado de esta operación un grado de similitud.

2.1.4 Módulos de utilidad

Al haber funciones compartidas entre los distintos módulos, se decidió crear unos módulos complementarios recogiendo toda esta funcionalidad común.

Estos dos módulos son colecciones de métodos auxiliares utilizados por los tres módulos principales. Uno de los dos ofrece una capa de abstracción para hacer todo tipo de operaciones con la base de datos. El otro contiene toda una serie de métodos dedicados al análisis de texto y comparación de texto.

Como se ha mencionado antes, se busca que los módulos sean lo más independientes posibles los unos de los otros y con el menor número posible de dependencias de librerías externas. Siendo este un objetivo fundamental para poder conseguir una mantenibilidad menos costosa se ha decidido crear una dependencia interna común en todos estos módulos. Esta dependencia consiste en un par de módulos con métodos de utilidad. Por dar un ejemplo, si en el futuro la librería externa *pymongo* cambia sus métodos de acceso a la base de datos, bastará con arreglar los métodos comunes definidos en estos módulos de utilidad para que los módulos A, B y C

continúen funcionando como se espera. Si no se hicieran estos módulos de utilidad, habría que arreglar el problema tres veces (una por modulo) en vez de una.

2.1.5 Requisitos no funcionales comunes a todos los módulos

RNF1. Todo el código será open source.

El trabajo tiene como finalidad ser una base para posteriores trabajos, por lo que es fundamental que el código de este trabajo sea abierto.

RNF2. El trabajo estará enfocado a ser usado por otros desarrolladores/estudiantes.

Todo el trabajo y en específico el módulo C debe de estar enfocado en ser tomado como relevo en futuros trabajos relacionados con el análisis de noticias y medios de comunicación.

2.2 Diseño de la base de datos

Como se deja ver en la Figura 3-1 y se menciona en los diseños de los módulos A y B, la base de datos tiene dos colecciones con el siguiente formato:

- **Artículos**. Cuyos documentos tendrán los siguientes campos:
  - (String) ***URL***.
    Extraída del feed RSS.
  - (String) **Titulo**.
    Extraído del procesamiento de la página web efectuado por la librería newspaper3k.
  - (String) **Autor**.
    Autor o autores si son mencionados o nombre de la empresa en su defecto.
  - (String) **Cuerpo**.
    Texto extraído del procesamiento de la página web.
  - (*ISODate*) **Fecha de publicación**.
    Fecha y hora de publicación extraídas del feed RSS o del procesamiento de la página web.
  - (String) **Palabras Clave**.
    Obtenidas gracias al módulo de procesamiento de lenguaje natural de la librería newspaper3k aplicándolo al campo Cuerpo.

  Como característica especial y clave para el funcionamiento del módulo B, los campos *Titulo*, *Cuerpo* y *Palabras Clave* son tratados como índices de texto *MongoDB*.

- **Eventos.** Cuyos documentos tendrán el siguiente campo:
    - (Array de strings) **Artículos**.
        - Consistirá de un array de URLs de artículos que hablan del mismo evento. Al no haber dos artículos con la misma URL dentro de la base de datos, estas URLs serán utilizadas como clave para poder buscar los artículos en la colección de estos.

3. Desarrollo

3.1 Tecnologías utilizadas

3.1.1 Base de datos: MongoDB

La base de datos elegida para la implementación del proyecto es MongoDB en su versión 3.2 o superior.

La elección de esta base de datos no relacional viene dada por una de las características que ofrece desde su versión MongoDB 2.4. Esta es la funcionalidad de los índices de texto (o texto indexado) y las consultas de texto basadas en ellos. Ambos permiten buscar con relativa inmediatez documentos que contengan las palabras indicadas en la consulta.

Esta funcionalidad es clave para poder implementar una solución rápida al objetivo que se tiene de poder agrupar artículos que cubran un mismo evento. Se requiere la versión especifica de MongoDB 3.2 ya que es la primera en incluir la versión 3 de índices y consultas de texto.

3.1.2 Lenguaje de programación: Python

El lenguaje de programación elegido es *Python*. La herramienta es utilizada para la obtención y análisis de artículos periodísticos. Es necesario un lenguaje con una gran comunidad y que ofrezca una amplia variedad de librerías y *APIs* para los distintos módulos en los que la herramienta está dividida.

En específico se usa la versión *Python 3.7* ya que esta utiliza el formato de codificación de caracteres *UTF-8* en vez de limitarse a *ASCII* como lo hacen las anteriores versiones. Esto resulta interesante por la naturaleza del trabajo; el análisis de texto. Por acotar el trabajo de fin de grado, este se ha hecho solo enfocado a noticias en español. Pero el hecho de utilizar codificación *UTF-8* deja abierta la posibilidad de adaptar con mayor facilidad la herramienta al análisis en otros idiomas.

Pero no solo esta característica es importante. Uno de los principales objetivos, no de la herramienta que se ha implementado sino subyacente al tema del trabajo, es que este sea la base para poder implementar un sistema con mayor funcionalidad de análisis, que sea base para poder hacer un mejor estudio en trabajos posteriores. Es por ello por lo que la versión utilizada de *Python* sea la 3.X en vez de la versión 2.X. Sería un error utilizar esta última en un proyecto

nuevo dada la noticia de la discontinuidad de mantenimiento de la versión 2.X el 1 de enero de 2020.

Otra razón, siguiendo el razonamiento de constituir base para trabajos posteriores, es el interés que puede tener en que esto derive en trabajos sobre *Machine Learning* o *Data Science*. Por ejemplo, para entrenar modelos de predicción de relaciones entre noticias, análisis de estilo de los autores, entrenamiento de nuevos modelos para motores de procesamiento del lenguaje natural. La comunidad más grande hoy en día en estos temas y que desarrolla las librerías más potentes es la comunidad de *Python*.

3.2 Librerías

Se ha intentado en mayor medida evitar el uso de librerías externas para hacer este proyecto más fácil de mantener. A pesar de esto, por limitación de tiempo o por cuestiones prácticas, se ha hecho uso de las siguientes librerías de Python:

3.2.1 Pymongo

Paquete de Python que implementa una API para interactuar con MongoDB. Es una librería utilizada y mantenida por una gran comunidad. Esta no debería suponer ningún problema en cuanto a mantenibilidad del sistema creado en este trabajo.

3.2.2 Feedparser

Librería de Python para la descarga y procesado de feeds RSS. Es un paquete bien mantenido y muy bien documentado. RSS es un formato standard basado en XML. Este formato tiene múltiples versiones, todas ellas pueden ser procesadas por esta librería.

Por ser RSS un standard y feedparser un paquete tan bien mantenido y documentado que parece un built-in de Python, estimo que el uso de esta librería añadiéndola como dependencia del módulo A, no supondrá ningún problema a la hora de mantener el sistema hecho en este trabajo.

3.2.3 Newspaper3k

Es un paquete de Python que proporciona una librería de funcionalidades para extraer artículos de noticias directamente de las páginas web de los medios de comunicación. Es capaz de extraer títulos, cuerpos, imágenes, videos, cabeceras… partiendo del código HTML de la página web. También proporciona métodos de procesado de lenguaje natural para la creación de resúmenes y obtención de palabras clave.

Este paquete puede resultar una dependencia problemática ya que la extracción de información la hace basándose en heurísticas. Ha habido en el pasado hilos de discusión abiertos en el repositorio inicial reportando que la extracción no resultaba efectiva en webs específicas o nuevas. Al tener una comunidad de tamaño medio estos problemas se resuelven relativamente rápido, pero sabiendo también como de rápido evolucionan las tecnologías cuando hablamos del mundo frontend, podemos suponer que en páginas de noticias nuevas no funcione como se espera al principio y por un tiempo.

A pesar de esto, durante el tiempo de desarrollo y pruebas, no he encontrado medio español del que no pudiera extraer correctamente los campos de su web.

3.2.4 Textdistance

Textdistance es un paquete de Python que proporciona más de treinta algoritmos distintos para la comparación de textos. Estos algoritmos están enfocados a obtener la similitud desde diferentes puntos de vista; similitud basada en edición, en división de los textos en subcadenas, basados en secuencias de palabras/frases, basados en compresión, basados en la fonética del texto, etc.

En Python hay ya muchas librerías dedicadas a esta tarea. La elección de esta viene dada por el hecho de que está completamente hecha en lenguaje Python haciendo solamente uso de los métodos built-in del lenguaje. Esto hace que los métodos no sean tan rápidos comparados con los de otras librerías que utilizan Cython o módulos compilados en C++. Pero a su vez el hecho de que la librería esté hecha puramente en Python y sin hacer uso de librerías externas hace que esta tenga un coste de mantenimiento prácticamente nulo. Esto hace que el uso de esta dependencia en el código de este trabajo no tenga que preocuparse en términos de mantenibilidad de este paquete.

3.2.5 Spacy

Este paquete de Python proporciona la librería más potente de procesamiento de lenguaje natural. Ha sido incluida en el proyecto debido a su magnitud y comunidad. Es un paquete muy bien mantenido y que ofrece modelos para análisis de texto en español. Uno de ellos y el que ha sido utilizado es su tagger. Un módulo que analiza un texto sintácticamente identificando palabra a palabra su tipo (verbo, articulo, preposición, etc.). Ha sido utilizado para la obtención de nombres propios en los títulos y cuerpos de los artículos de noticias.

3.2.6 Matplotlib

Es la librería más popular de Python para el dibujo de graficas en dos dimensiones. Esta librería tiene muy buena documentación y está bien mantenida. Es a su vez base de otras librerías de dibujo de graficas más complejas.

3.2.7 ETE3

Usando como base la librería matplotlib, ete3 es actualmente la librería más completa para la representación de árboles. Como matplotlib, esta librería tiene unas páginas de documentación muy extensas que proporcionan ejemplos prácticos de todo lo que es posible hacer con ella.

Este paquete ha sido utilizado para el dibujo automático de árboles genealógicos que representan la similitud entre artículos que hablan de un mismo evento.

3.3 Implementación

Lo primero que se implementó fueron los módulos de utilidad, en específico el dedicado a operaciones con la base de datos. Una vez hechos y probados fueron utilizados por los diferentes módulos principales.

3.3.1 Módulo A: populate_db.py

Haciendo uso del módulo de utilidad lo primero que se hizo fue asegurarse de que cada vez que se ejecute el sistema, el módulo A limpiara la base de datos borrándola.

Para hacer un control exacto de las URLs procesadas se ha elegido usar un set() de Python, el cual no permite tener elementos repetidos en él y la comprobación de si un elemento existe dentro del set es de poco costo computacional.

Para habilitar la búsqueda basada en índices de texto el módulo A crea la colección de artículos configurando los campos descritos en la sección 2.2 de este documento como índices de texto MongoDB.

El módulo leerá las URL a los feed RSS contenidos en el archivo de texto de entrada usando la librería feedparser. Por cada archivo XML de los feed RSS obtendrá las URLs a los artículos que serán añadidos al set mencionado.

Por cada URL en el set, el módulo, haciendo uso de la librería newspaper3k obtendrá todos los campos relativos al artículo y descritos en la sección 2.2 de este documento.

3.3.2 Módulo B: group_by_event.py

Por cada artículo de la base de datos que se encuentra en la colección de artículos:

Se seleccionan los nombres propios con mayor peso en el artículo usando el tagger español de la librería spacy.

Estos nombres propios se concatenan con las palabras clave obtenidas en el módulo anterior por la librería newspaper3k.

Con esta string se procede a hacer una búsqueda de texto basada en índices y los resultados se ordenan por puntuación.

Resultado a resultado se analiza si el artículo realmente habla del mismo evento que el artículo utilizado de referencia.

Cada una de las URLs de los artículos que se detecta que hay una gran probabilidad de que hablen del mismo evento se almacenan en la base de datos en la colección creada por este módulo de eventos.

3.3.3. Módulo C: analyse_events.py

Este módulo hace una búsqueda en la base de datos sobre los eventos más populares, usando como medida de popularidad la cantidad de artículos que tratan el evento.

Con el evento se obtienen todas las URL de los artículos que hablan del mismo y haciendo búsquedas en la base de datos se obtienen en memoria principal toda la información relativa a los artículos.

Teniendo los artículos en memoria, se calcula el grado de similitud con la librería textdistance.

Basándose en los grados de similitud, el módulo pinta un árbol genealógico en el que se pueden ver las noticias que tienen una fuente en común.

3.3.4. Módulos de utilidad

Estos módulos se han dejado fuera del esquema del diseño. No son más que una colección de métodos que actúan como interfaz para interactuar, a un mayor nivel de abstracción, con la librería pymongo. Además, ofrecen una serie de métodos para el análisis y comparación de textos usando la librería anteriormente mencionada textdistance.

3.3.4.1 db_utils.py

Este módulo contiene todos los métodos relativos a la interacción con la base de datos. Además, sirve como archivo de configuración en el que se pueden cambiar parámetros como el nombre de las colecciones a usarse, el idioma de los textos para así crear los índices de texto, el nombre de la base de datos y la información necesaria relativa a la conexión con la base de datos (Puerto, dirección IP, etc.)

3.3.4.2 analysis_utils.py

Este módulo contiene todos los métodos relativos al análisis de texto y debe ser tomado como base para ser ampliado en futuros trabajos.

4. Pruebas y resultado

Las pruebas sobre la funcionalidad del código se han hecho por etapas según eran necesarias para probar un correcto funcionamiento de la aplicación.

En particular se han hecho una serie de tests mientras se desarrollaba, a modo de prueba y error, parar estimar los siguientes parámetros:

- Los pesos que utilizar cuando se crean los índices de texto en la colección de artículos de la base de datos MongoDB. Se terminó concluyendo que hay que dar un mayor peso al campo título para obtener unas puntuaciones obtenidas en búsquedas más coherentes y efectivas a la hora de determinar si un artículo habla del mismo evento que el artículo de referencia.
- A su vez se estimó, con multitud de pruebas, la cantidad media de nombres propios comunes entre artículos que hablan de un mismo evento. Esto afinó de nuevo la eficacia de obtención de artículos para la creación de eventos en la base de datos.

Ambos factores fueron obtenidos teniendo en cuenta que no es aceptable dejar fuera de un evento un artículo que hable del mismo.

El resultado de este trabajo es una base sobre la que partir para poder desarrollar trabajos centrados en el tratamiento de la información de artículos de noticias y no tanto en la obtención de estos. El análisis de fuentes comunes entre artículos representado como árbol genealógico constituye una pequeña prueba de concepto de lo mucho que puede hacerse teniendo la información necesaria ya clasificada por evento. En la siguiente sección se hablará de todas aquellas mejoras que se pueden llevar a cabo para hacer un sistema más completo y autónomo, así como ideas para trabajos futuros.

5. Conclusiones y trabajo futuro

5.1 Conclusiones

Sabiendo lo razonable que es hoy en día dudar de la fuente de una noticia y sus intenciones, concluyo que se ha de empezar a utilizar todas las herramientas de las que disponemos para poder analizar y entender los intereses que mueven a los medios, para poder discernir entre el arte de informar y la manipulación.

El diseño y desarrollo ha llevado tiempo en la búsqueda de las tecnologías, librerías y herramientas necesarias para cumplir el objetivo del trabajo y hacer de este una base desde la que partir en trabajos futuros.

5.2. Trabajo futuro

Podemos clasificar el trabajo futuro en dos secciones:

- Mejoras sobre los **módulos A y B**. El módulo se podría modificar para usarse como Daemon que monitoriza los feeds RSS cada un determinado tiempo en busca de nuevos artículos incluidos y que ser clasificados por el módulo B. A su vez se puede modificar el módulo B para que esté escuchando todas aquellas nuevas inserciones de artículos en la base de datos para acto seguido clasificarlos en eventos.
  Ambas modificaciones tienen como objetivo crear un sistema que monitorice los medios y analice a tiempo real.
- Trabajos centrados en el **módulo C**. Como se repite una y otra vez a lo largo de este documento, el objetivo principal de este trabajo es ofrecer un software base para el estudio de relaciones entre artículos de noticia que habla sobre un mismo evento. Algunas cosas objeto de análisis y gran interés pueden ser por ejemplo:
  - Análisis del grado de objetividad y sentimiento de los artículos en español teniendo que entrenar modelos de Machine Learning propios.
  - Análisis semántico y búsqueda de contradicciones lógicas entre artículos.
  - Sistema para la detección de fake news basado en patrones.
  - Análisis y medida de la influencia política en los diferentes medios.

Referencias

[1] Benjamin D. Horne, Jeppe Nørregaard and Sibel Adal, “Different Spirals of Sameness: A Study of Content Sharing in Mainstream and Alternative Media” 2 April 2019.

[2] Benjamin D. Horne, Dorit Nevo, John O’Donovan, Jin-Hee Cho, and Sibel Adal “Rating Reliability and Bias in News Articles: Does AI Assistance Help Everyone?” 2 April 2019.

[3] Cristian Popa, Alexandru Popa “NewsCompare - a novel application for detecting news influence in a country” 1 April 2019

[4] Kai Shu, Deepak Mahudeswaran, SuhangWang, and Huan Liu “Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation” 21 March 2019

[5] Jooyeon Kim, Dongkwan Kim, and Alice Oh “Homogeneity-Based Transmissive Process To Model True and False News in Social Networks” 16 November 2018

[6] Dorje C. Brody1 and David M. Meier “How to model fake news” 26 October 2018

[7] Benjamin D. Horne, William Dron, Sibel Adal “Models for Predicting Community-Specific Interest in News Articles” August 2018

[8] Abbas Ehsanfar, Mo Mansouri “Incentivizing the Dissemination of Truth Versus Fake News in Social Networks” June 2017

[9] Amin Khajehnejad, Shima Hajimirza “A Bayesian Model for False Information Belief Impact, Optimal Design, and Fake News Containment” 13 March 2018

[10] Zilong Zhao, Jichang Zhao, Yukie Sano, Orr levy, Hideki Takayasu, Misako Takayasu, Daqing Li, Shlomo Havlin “Fake news propagate differently from real news even at early stages of spreading” March 2018.

Glosario

API	Application Programming Interface.
ASCII	American Standard Code for Information Interchange.
Built-in	Refiriéndose a funcionalidad concreta de un lenguaje, expresa que esta tiene dada por la implementación del propio lenguaje.
Cython	Lenguaje de programación o extensión de Python para hacer posible la creación de módulos en C y C++.
C++	Lenguaje de programación.
Data Science	Campo que estudia la extracción y entendiendo de la relación entre datos.
ete3	Librería de Python para el dibujo de árboles.
Fake news	Noticias falsas. Herramienta para la manipulación de masas.
Frontend	Referido a la capa de presentación.
feedparser	Librería Python para el proceso de feeds RSS.
Feed RSS	Archivo generado a partir del contenido de una página web a modo de sumario del mismo. Normalmente en formato XML.
HTML	HyperText Markup Language
ISOdate	Formato estándar de representación de fechas.
IP	Internet Protocol address.
Machine Learning	Aprendizaje automático o aprendizaje de máquina
matplotlib	Librería Python para el dibujo de graficas.
MongoDB	Base de datos no relacional y de código abierto.
newspaper3k	Librería Python para el procesamiento y extracción de datos de artículos periodísticos en la web.
Parser	Aplicación dedicada procesar datos en un formato determinado.
pymongo	Librería Python para la integración de bases de datos MongoDB.
Python	Lenguaje de programación.
set()	Estructura de datos cuya particularidad es no permitir repetición de datos.
spacy	Librería Python para el procesamiento del lenguaje natural.
String	Tipo de datos para la representación de cadenas de texto.

Tagger	En el procesamiento de lenguaje natural es un método para clasificar palabras usando un modelo entrenado en analizar sintácticamente texto.
textdistance	Librería de Python para la comparación de textos y obtención de grado de similitud o distancia.
UNIX	Familia de sistemas operativos evolucionados y creados a partir de un antecesor común.
URL	Uniform Resource Locator.
UTF-8	8-bit Unicode Transformation Format.
XML	eXtensible Markup Language.

Anexos

A. Manual de instalación y uso

Para obtener el proyecto:

```
git clone https://alvarocortesd@bitbucket.org/alvarocortes/tfg.git
```

Para poder hacer uso del código hace falta instalar *MongoDB* en la maquina a utilizar.
Una vez instalado habrá que instalar *Python 3* en el sistema.
Con ambos instalados ya puede hacerse uso del *Makefile* incluido en el repositorio para instalar las dependencias de *Python* utilizadas:

```
make install
```

Para iniciar la base de datos con el comando:

```
make start_db
```

Una vez iniciado la base de datos se podrá ejecutar el siguiente comando para ejecutar la aplicación completa y ver por defecto un pequeño análisis en la consola y la representación del árbol de los artículos del evento más popular:

```
make run
```